




- Introduction to SQL
- Basic structure of SQL commands
- Data Definition
- Data Manipulation
- Aggregation


Introduction to SQL

- SQL is a transform-oriented language with two major components. These are the DDL for defining the database structure and the DML for retrieving and updating data.




Introduction to SQL

- SQL does not contain flow control commands. These must be implemented using a programming or job-control language, or interactively by the decisions of the user.




Introduction to SQL

- SQL is relatively easy to learn. SQL is a nonprocedural language,
- you specify what information you require, rather than how to get it.



Introduction to SQL

- An ISO standard now exists for SQL, making it both the formal and de facto standard language for relational databases.
- The most popular and widely implemented is referred to as SQL2 or SQL/92.



Basic structure of SQL commands

- SQL statement consists of reserved words and user-defined words.
- Reserved words are a fixed part of SQL and must be spelt exactly as required and cannot be split across lines.
- User-defined words are made up by user and represent names of various database objects such as relations, columns and views.

Basic structure of SQL commands

- Most components of an SQL statement are case insensitive
- SQL statements are more readable with indentation and lineation.

Data Manipulation

- A query in SQL can consist of up to six clauses, but only the first two are mandatory.

```
SELECT [DISTINCT | ALL]
{* | [column_expression [AS new_name]] [,... ] }
```

Data Manipulation

```
FROM table_name [alias] [, ...]
[WHERE condition]
[GROUP BY column_list]
[HAVING condition]
[ORDER BY column_list]
```

Data Manipulation CONT:

- SELECT specifies which columns are to appear in output.
- FROM specifies table(s) to be used.
- WHERE filters rows.

Data Manipulation CONT:

- GROUP BY forms groups of rows with same column value.
- HAVING filters groups subject to some condition.
- ORDER BY specifies the order of the output.

MANIPULATION EXAMPLES

- Specific Columns, Specific Rows.
- Find the salaries of employees named Brown.
 - ```
SELECT Salary as Remuneration
FROM Employee
WHERE Surname = 'Brown';
```

## MANIPULATION EXAMPLES

All Columns, Specific Rows.

- Find all the information relating to employees named Brown.
- SELECT \*  
FROM Employee  
WHERE Surname = 'BROWN';

## MANIPULATION EXAMPLES

Attribute expressions (calculated field)

- Find the monthly salary of the employees named White.
- SELECT Salary / 12 as MonthlySalary  
FROM Employee  
WHERE Surname = 'White';

## MANIPULATION EXAMPLES

- Simple join query
  - Find the names of the employees and the cities in which they work.
- SELECT Employee.FirstName,  
Employee.Surname, Department.City  
FROM Employee, Department  
WHERE Employee.Dept =  
Department.DeptName;

## MANIPULATION EXAMPLES

- Using predicate disjunction
  - Find the first names and surnames of the employees who work in either the Administration or the Production department.
- SELECT FirstName, Surname  
FROM Employee  
WHERE Dept = 'Administration' OR  
Dept = 'Production';

## MANIPULATION EXAMPLES

- Using complex logical expression
  - Find the first names of the employees named Brown who work in the Administration department or the Production department.
- SELECT FirstName  
FROM Employee  
WHERE Surname = 'Brown' AND  
(Dept = 'Administration' OR Dept =  
'Production');

## MANIPULATION EXAMPLES

- Using the pattern match search condition (like/not like)
  - Find the employees with surnames that have 'r' as the second letter and end in 'n'.
- SELECT \*  
FROM Employee  
WHERE Surname LIKE '\_r%n';

## MANIPULATION EXAMPLES

- Null search condition
  - List the details of all viewings on property PG4 where a comment has not been supplied
- SELECT viewing.pno, viewing.rno, Date  
FROM viewing  
WHERE pno='PG4' AND comment IS NULL;

## MANIPULATION EXAMPLES

- Q11: Sorting results
  - Produce an abbreviated list of properties arranged in order of property type.
- SELECT Pno, Type, Rooms, Rent  
FROM Property  
ORDER BY Type;

## MANIPULATION EXAMPLES

- Comparison search condition
  - List all staff with a salary greater than 10,000.
- SELECT Staff.Sno, Staff.FName,  
Staff.LName, Position, Salary  
FROM Staff  
WHERE Salary > 10000;

## MANIPULATION EXAMPLES

- Range search condition
  - List all staff with a salary between 20,000 and 30,000.
- SELECT staff.Sno, staff.FName,  
staff.LName, staff.Position, staff.Salary  
FROM staff  
WHERE staff.Salary BETWEEN 20000  
AND 30000;

## MANIPULATION EXAMPLES

- Set membership search condition
  - List all Managers and Deputy Managers.
- SELECT staff.Sno, staff.FName,  
staff.LName, staff.Position  
FROM staff  
WHERE position in ('Manager',  
'Deputy');

## Aggregation

- ISO standard defines five aggregate functions. These are:
  - COUNT returns number of values in a specified column.
  - SUM returns sum of values in a specified column.

## Aggregation

- AVG returns average of values in a specified column.
- MIN returns smallest value in a specified column.
- MAX returns largest value in a specified column.

## Aggregation examples

- Q16: The Count function
  - How many properties cost more than £350 per month for rent?
- SELECT Count(\*) AS count  
FROM property  
WHERE property.Rent > 350;

## Aggregation examples

- The Max, Min and Avg function
  - Find the minimum, maximum and average staff salary.
- SELECT MIN(salary) AS MIN,  
MAX(salary) AS MAX, AVG(salary) AS  
AVG  
FROM staff;

## Aggregation examples

- Using the Group By clause
  - Find the number of staff working in each branch and the total of their salaries.
- SELECT bno, COUNT(sno) AS count,  
SUM(salary) AS sum  
FROM staff  
GROUP BY bno  
ORDER BY bno;

## Aggregation examples

- Using predicates on grouping results
  - For each branch office with more than one member of staff, find the number of staff working in each branch and the sum of their salaries.

## Aggregation examples

- SELECT bno, COUNT(sno) AS count,  
SUM(salary) AS sum  
FROM staff  
GROUP BY bno  
HAVING COUNT(SNO) > 1;

## Multiple Grouping Columns

- Find number of properties handled by each staff member.
- ```
SELECT s.branchNo, s.staffNo, COUNT(*) AS count
FROM Staff s, PropertyForRent p
WHERE s.staffNo = p.staffNo
GROUP BY s.branchNo, s.staffNo
ORDER BY s.branchNo, s.staffNo;
```

